Amendments to the Claims:

This listing of claims replaces all prior versions and listings of claims in the application.

Listing of Claims:

1. (Currently Amended) A method for dynamically programming ~~Field Programmable Gate Arrays~~ a field programmable gate array (FPGA) in a coprocessor, ~~the coprocessor coupled to a processor,~~ the method comprising:

providing a processor and a coprocessor that is separate from the processor, the coprocessor being coupled to the processor and including a field programmable gate array (FPGA);

~~(a) beginning an execution of~~ executing an application ~~by~~ using the processor;

~~(b)~~ the coprocessor receiving an instruction from the processor ~~by the coprocessor~~ to perform a function for the application;

~~(c)~~ determining that the field programmable gate array (FPGA) in the coprocessor is not programmed ~~with logic for~~ to perform the function for the application;

~~(d)~~ fetching a configuration bit stream ~~for~~ associated with the function for the application responsive to the determination that the field programmable gate array (FPGA) is not programmed to perform the function for the application; and

~~(e)~~ dynamically programming the field programmable gate array (FPGA) in accordance with the configuration bit stream to perform the function for the application.

2. (Currently Amended) The method of claim 1, wherein the ~~issuing step (b)~~ coprocessor further comprises ~~(b1)~~ an Auxiliary Processing Unit (APU) interface ~~between the processor and the coprocessor~~ for receiving instructions from the processor, the Auxiliary Processing Unit

(APU) interface determining whether a given instruction is to be processed by the coprocessor.

3. (Currently Amended) The method of claim 2, wherein the Auxiliary Processing Unit (APU) interface determining whether a given instruction is to be processed by the coprocessor includes ~~step (c) comprises: (c1) determining that~~ the Auxiliary Processing Unit (APU) interface ~~issued~~ issuing a faulty commit if the given instruction is to be processed by the coprocessor and the field programmable gate array (FPGA) in the coprocessor is not programmed to perform a function corresponding to the given instruction.

4. (Currently Amended) The method of claim [[1]] 3, ~~wherein the fetching step (d) comprises: (d1)~~ further comprising:

the processor initiating an exception subroutine ~~by the processor~~ responsive to the Auxiliary Processing Unit (APU) interface issuing a faulty commit; and

~~(d2)~~ the exception routine identifying ~~fetching~~ the configuration bit stream ~~for~~ associated with the function ~~by an exception subroutine of the processor~~ for the application.

5. (Currently Amended) The method of claim 4, wherein the ~~initiating step (d1)~~ the processor initiating an exception subroutine comprises the processor ~~: (d1i) determining that the coprocessor issued a faulty commit; and (d1ii)~~ branching to the exception subroutine ~~by the processor~~ in response to the faulty commit.

6. (Currently Amended) The method of claim 4, wherein the exception routine identifying the configuration bit stream ~~fetching step (d2)~~ comprises:

~~(d2i)~~ the exception routine decoding a function identifier ~~by the exception subroutine~~

associated with the function corresponding to the given instruction;

~~(d2ii) requesting and being granted ownership of the function;~~

~~(d2iii)~~ the processor fetching the configuration bit stream ~~for~~ associated with the function corresponding to the given instruction from a memory; and

~~(d2iv) identifying an exception type and a coprocessor instruction type for the configuration bit stream; and~~

~~(d2v)~~ the processor sending the configuration bit stream associated with the function corresponding to the given instruction to a direct memory access (DMA) channel coupled to the coprocessor for programming the field programmable gate array (FPGA) in the coprocessor.

7. (Currently Amended) The method of claim 1, wherein ~~the~~ dynamically programming ~~step (e)~~ the field programmable gate array (FPGA) comprises: ~~(e1)~~ the processor performing a sequence of load and store instructions ~~by~~ in accordance with an exception subroutine of the processor to program the field programmable gate array (FPGA) in accordance with the configuration bit stream.

8. (Currently Amended) The method of claim 1, further comprising: ~~(f)~~ the processor reissuing ~~receiving a reissuance of~~ the instruction ~~by~~ to the coprocessor responsive to the field programmable gate array (FPGA) being dynamically reconfigured to perform the function for the application.

9. (Currently Amended) A method for dynamically programming ~~Field Programmable Gate Arrays~~ a field programmable gate array (FPGA) in a coprocessor, ~~the coprocessor coupled to a processor,~~ the method comprising:

providing a processor and a coprocessor that is separate from the processor, the coprocessor being coupled to the processor and including a field programmable gate array (FPGA) and an Auxiliary Processing Unit (APU) interface;

~~(a) beginning an execution of~~ executing an application ~~by~~ using the processor;

~~(b)~~ the coprocessor receiving an instruction from the processor ~~by the coprocessor~~ to perform a function for the application;

~~(c)~~ the Auxiliary Processing Unit (APU) interface issuing a faulty commit when the field programmable gate array (FPGA) in the coprocessor is not programmed ~~with logic for~~ to perform the function;

~~(d)~~ the processor initiating an exception subroutine ~~by the processor~~ in response to the faulty commit;

~~(e)~~ the exception subroutine fetching a configuration bit stream ~~for~~ associated with the function ~~by the exception subroutine~~ for the application; and

~~(f)~~ the exception subroutine performing a sequence of load and store instructions ~~by the exception subroutine~~ to program the field programmable gate array (FPGA) in accordance with the configuration bit stream to perform the function.

10. (Currently Amended) A computer readable medium with program instruction tangibly stored thereon for dynamically programming ~~Field Programmable Gate Arrays~~ a field programmable gate array (FPGA) in a coprocessor, the coprocessor being coupled to a processor that is separate from the coprocessor, the computer readable medium comprising ~~the~~ instructions for:

~~(a) beginning an execution of~~ executing an application ~~by~~ using the processor;

~~(b)~~ the coprocessor receiving an instruction from the processor ~~by the coprocessor~~ to

perform a function for the application;

~~(c)~~ determining that the field programmable gate array (FPGA) in the coprocessor is not programmed ~~with logic for~~ to perform the function for the application;

~~(d)~~ fetching a configuration bit stream ~~for~~ associated with the function for the application; and

~~(e)~~ programming the field programmable gate array (FPGA) in accordance with the configuration bit stream to perform the function for the application.

11. (Currently Amended) The computer readable medium of claim 10, wherein the coprocessor further includes an Auxiliary Processing Unit (APU) interface for receiving instructions from the processor, and the computer readable medium further ~~issuing instruction (b)~~ comprises ~~the~~ instructions for: ~~(b1) receiving the instruction by an~~ the Auxiliary Processing Unit (APU) interface ~~between the processor and the coprocessor~~ determining whether a given instruction is to be processed by the coprocessor.

12. (Currently Amended) The computer readable medium of claim 11, wherein the ~~determining instruction (c)~~ instructions for the Auxiliary Processing Unit (APU) interface determining whether a given instruction is to be processed by the coprocessor comprises ~~the~~ instructions for: ~~(c1) determining that~~ the Auxiliary Processing Unit (APU) interface ~~issued~~ issuing a faulty commit if the given instruction is to be processed by the coprocessor and the field programmable gate array (FPGA) in the coprocessor is not programmed to perform a function corresponding to the given instruction.

13. (Currently Amended) The computer readable medium of claim [[10]] 12, ~~wherein the~~

~~fetching instruction (d) comprises the~~ further comprising instructions for:

~~(d1)~~ the processor initiating an exception subroutine ~~by the processor~~ responsive to the Auxiliary Processing Unit (APU) interface issuing a faulty commit; and

~~(d2)~~ the exception routine identifying ~~fetching~~ the configuration bit stream ~~for~~ associated with the function ~~by an exception subroutine of the processor~~ for the application.

14. (Currently Amended) The computer readable medium of claim 13, wherein the instructions for processor initiating an exception subroutine ~~instruction (d1)~~ comprises ~~the~~ instructions for: ~~(d1i) determining that the coprocessor issued a faulty commit; and (d1ii)~~ the processor branching to the exception subroutine ~~by the processor~~ in response to the faulty commit.

15. (Currently Amended) The computer readable medium of claim 13, wherein the ~~fetching instruction (d2)~~ instructions for the exception routine identifying the configuration bit stream comprises ~~the~~ instructions for:

~~(d2i)~~ the exception routine decoding a function identifier ~~by the exception subroutine~~ associated with the function corresponding to the given instruction;

~~(d2ii) requesting and being granted ownership of the function;~~

~~(d2iii)~~ the processor fetching the configuration bit stream ~~for~~ associated with the function corresponding to the given instruction from a memory; and

~~(d2iv) identifying an exception type and a coprocessor instruction type for the configuration bit stream; and~~

~~(d2v)~~ the processor sending the configuration bit stream associated with the function corresponding to the given instruction to a direct memory access (DMA) channel coupled to the

coprocessor for programming the field programmable gate array (FPGA) in the coprocessor.

16. (Currently Amended) The computer readable medium of claim 10, wherein the instructions for dynamically programming ~~instruction (e)~~ the field programmable gate array (FPGA) comprises ~~the~~ instructions for: ~~(e1)~~ the processor performing a sequence of load and store instructions ~~by~~ in accordance with an exception subroutine of the processor to program the field programmable gate array (FPGA) in accordance with the configuration bit stream.

17. (Currently Amended) The computer readable medium of claim 10, further comprising ~~the~~ instructions for: ~~(f)~~ the processor reissuing ~~receiving a reissuance of~~ the instruction ~~by~~ to the coprocessor responsive to the field programmable gate array (FPGA) being dynamically reconfigured to perform the function for the application.

18. (Currently Amended) A computer readable medium with programming instructions tangibly stored thereon for dynamically programming ~~Field Programmable Gate Arrays~~ a field programmable gate array (FPGA) in a coprocessor, the coprocessor including an Auxiliary Processing Unit (APU) interface and being coupled to a processor that is separate from the coprocessor, the computer readable medium comprising ~~the~~ instructions for:

~~(a) beginning an execution of~~ executing an application ~~by~~ using the processor;

~~(b)~~ the coprocessor receiving an instruction from the processor ~~by the coprocessor~~ to perform a function for the application;

~~(c)~~ the Auxiliary Processing Unit (APU) interface issuing a faulty commit when the field programmable gate array (FPGA) in the coprocessor is not programmed ~~with logic for~~ to perform the function;

~~(d)~~ the processor initiating an exception subroutine ~~by the processor~~ in response to the faulty commit;

~~(e)~~ the exception subroutine fetching a configuration bit stream ~~for~~ associated with the function ~~by the exception subroutine~~ for the application; and

~~(f)~~ the exception subroutine performing a sequence of load and store instructions ~~by the exception subroutine~~ to program the field programmable gate array (FPGA) in accordance with the configuration bit stream to perform the function.

19. (Cancelled)

20 (New) A system comprising:

a program memory;

a plurality of processors in communication with the program memory, each processor sharing the program memory among other ones of the plurality of processors;

a plurality of coprocessors separate from the plurality of processors, each coprocessor being coupled to a corresponding processor and including a field programmable gate array (FPGA); and

a shared resource manager operable to program each field programmable gate array associated with the plurality of coprocessors,

wherein a first processor of the plurality of processors is operable to execute an application and send an instruction from the program memory to each of the plurality of coprocessors to perform a function for the application,

if none of the field programmable gate array (FPGAs) associated with the plurality of coprocessors are programmed to perform the function for the application, then the shared

resource manager is operable to dynamically program any one of the field programmable gate arrays (FPGAs) to perform the function for the application, in which the field programmable gate array (FPGA) selected to be dynamically programmed is selected in accordance with a least recently used algorithm, the least recently used algorithm specifying a function that can be disabled to free up logic resources within the field programmable gate array (FPGA) selected to be dynamically programmed.